

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
6 December 2001 (06.12.2001)

PCT

(10) International Publication Number
WO 01/93514 A1(51) International Patent Classification: H04L 12/58,
H04Q 7/2213, S-186 43 Vallentuna (SE) ERIKSSON, Henrik; Svart-
bäcksgatan 80 A, S-753 35 Uppsala (SE).

(21) International Application Number: PCT/SE01/01094

(74) Agent: DR. LUDWIG BRANN PATENTBYRÅ AB;
Box 17192, S-104 62 Stockholm (SE).

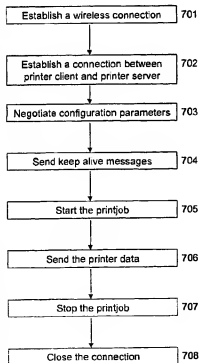
(22) International Filing Date: 17 May 2001 (17.05.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0002048-7 31 May 2000 (31.05.2000) SE(81) Designated States (national): AE, AG, AL, AM, AT, AU,
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU,
CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH,
GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KR, KZ, LC,
LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW,
MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK,
SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.(71) Applicant: ALLGON AB (SE/SE); Antennvägen 6, S-187
80 Täby (SE).(84) Designated States (regional): ARIPO patent (GH, GM,
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European
patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,
IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF,
CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TO).

[Continued on next page]

(54) Title: METHOD IN A TELECOMMUNICATIONS NETWORK FOR CONNECTING A DEVICE WIRELESSLY TO AN
OTHER

(57) Abstract: The present invention relates to a method and an arrangement in a data communications system. The object of the invention is to achieve a wireless communication between a processing unit and a printer using a safe transmission and an increased transmission range compared to the infrared transmission. The solution, according to the invention is a way of printing a document in a data communications system using a protocol profiled for printing in the Bluetooth protocol architecture.

WO 01/93514 A1

METHOD IN A TELECOMMUNICATIONS NETWORK FOR CONNECTING
A DEVICE WIRELESSLY TO ANOTHER

5 FIELD OF INVENTION

The present invention relates to a method and an arrangement in a data communications system according to the preambles of the independent claims. More specifically it relates to a processing unit wirelessly connected to a printer. It further relates to printing a document by means of the printer, the printer being
10 controlled by the processing unit.

DESCRIPTION OF RELATED ART

15 Processing units, e.g. PC's requiring to print documents uses typically a printer. A processing unit and a printer are generally communicating with each other through cables. But communication disruption caused by wire breakage or inadequate securing of the cable ends, added cost of providing a reliable cable and reliable associated connectors, tangling of the cables and requirements of flexibility, etc.
20 leads to a requirement of replacing the cables.

A way of communicating, using an infrared link instead of a cable is shown in the American patent US 6,055,062, which discloses an electronic printer having an attached accessory unit. The accessory unit handles e.g. optional media (e.g. paper) supply units and optional media output. To communicate with the accessory unit,
25 the printer uses a two-ways infrared communications connection to the accessory unit to which it is immediately adjacent.

However the range of the infrared link is short, so that the distance between processing unit and the printer have to be less than a few meters and there must be a clear line of sight between them.

The so-called Bluetooth interface is an example of a modern radio interface, which was originally intended as replacement for cables between units. The term Bluetooth is in this disclosure used as an example of usage of short-range radio communication. By replacing the cables, the short-range radio technology provides a universal bridge to existing data networks, a peripheral interface, and a mechanism to form small private ad hoc groupings of connected devices away from fixed network infrastructures or connected to a fixed network infrastructure via a gateway. Designed to operate in a noisy frequency environment, the Bluetooth radio uses a fast acknowledgement and frequency hopping scheme to make the link robust. Bluetooth radio modules avoid interference from other signals by hopping to a new frequency after transmitting or receiving a data packet, as shown in Figure 1 wherein the X-axis represents the frequency f and the Y-axis represents the time t . Compared with other systems operating in the same frequency band, the Bluetooth radio typically hops faster and uses shorter radio packets. This makes Bluetooth radio more robust than other systems. Use of Forward Error Correction (FEC) limits the impact of random noise on long-distance links.

Bluetooth radio is a wireless communication technology using a frequency-hopping scheme in the unlicensed Industrial Scientific Medical (ISM) band at 2,4 GHz. A frequency hop transceiver is applied to combat interference and fading. A shaped, binary FM modulation is applied to minimise transceiver complexity. The gross data rate is 1Mb/s and Time-Division Duplex (TDD) scheme is used for full duplex transmission.

The Bluetooth protocol is a combination of circuit and packet switching. In Figure 1, S1 denotes one time slot, and P1 denotes a packet covering three time slots. A time slot is 0,625 ms long. Time slots can be reserved for synchronous packets. Each packet is normally transmitted in a different hop frequency. A packet normally covers a single slot, but can be extended to cover up to five slots. Bluetooth can support an asynchronous data channel, up to three simultaneous synchronous voice channels, or a channel with simultaneously supports

asynchronous data and synchronous voice. Each voice channel supports 64 kb/s synchronous (voice) link. The asynchronous channel can support an asymmetric link of maximally 721 kb/s in either direction while permitting 57,6 kb/s in the return direction, or a 432,6 kb/s symmetric link.

5

In figure 2, the different function blocks of a system using short-range radio transceivers such as Bluetooth are shown. A radio unit 201 is connected to a link control unit 202 providing the base band. The link control unit 202 is connected to the Central Processing Unit, called CPU, 203 providing the link management. The CPU is connected to the memory 204 providing software functions and consisting of two memory units: a SRAM 205 and a FLASH 206. The CPU 203 is connected to a host interface 207. A SRAM is a fast temporary memory. FLASH is a programmable ROM.

15

Two or more, up to eight Bluetooth units sharing the same channel form a piconet, i.e. a piconet is a collection of devices connected via Bluetooth technology in an ad hoc fashion. Within a piconet a Bluetooth unit can have either of two roles: master or slave. Within each piconet there may be one and only one master, and up to seven active slaves, i.e. a piconet starts with two connected devices, such as a portable PC and a cellular telephone, and may grow to eight connected devices. All Bluetooth devices are peer units and have identical implementations. Any Bluetooth unit can become master in a piconet. A master unit is the device in a piconet whose clock and hopping sequence are used to synchronise all other devices within the piconet. A slave unit is every device in a piconet that is not a master.

The communication within a piconet is organised such that the master polls each slave according to some polling scheme. Master-to-slave transmission always starts in an even-numbered time-slot while slave-to-master transmission always starts in an odd-numbered time slot. With one exception the slave is only allowed to

30

transmit after have been polled by the master. The slave then starts its transmission in a slave-to-master time slot immediately following the packet received from the master. The master may or may not include data in the packets used to poll the slave. The only exception to the above principle is that when a slave has an established Synchronous Connection Oriented (SCO) link, the slave is always allowed to transmit in the pre-allocated slave-to-master slot, even if not explicitly polled by the master in the preceding master-to slave slot. The term SCO-link will be disclosed in more details below. In a Bluetooth communications system there is no direct transmission between slaves in a piconet.

10

The Bluetooth protocol stack will be described, according to the specifications of the Bluetooth system. The protocol stack which is depicted in figure 3, includes two Bluetooth units 301 and 302. In the figure the physical layer and the data link layer are shown.

15

Baseband BB

The base band describes the digital signal processing part of the hardware, i.e. the Bluetooth link controller, which carries the Bluetooth protocols and other low-level link routines. The Baseband resides in the physical layer 301 and the data link layer 304. The baseband specification defines two link types: Synchronous Connection-Oriented (SCO) links and Asynchronous Connection-Less (ACL) links. SCO links support real-time voice traffic using reserved bandwidth. ACL links support best effort traffic.

25

Link Manager Protocol LMP

LMP handles messages used for link set-up, security and control. LMP is layered over the Baseband protocol and resides in the data link layer 304.

30 Logical Link Control and Adaptation layer Protocol, L2CAP

L2CAP is also layered over the Baseband protocol and resides in the data link layer 304. L2CAP provides connection oriented and connectionless data

services to upper layer protocols with multiplexing capability, segmentation and reassemble operation, and group abstractions. The L2CAP Specification is only defined for ACL links.

5 Network layer 305

The network layer is currently not specified in the Bluetooth standard.

High level protocol or application 306

Device information, services and the characteristics of the services can be queried using the Service Discovery Protocol SDP. Like SDP, RFCOMM is layered on top of the L2CAP. RFCOMM is the 'cable replacement' protocol, which provides transport capabilities for high-level services (e.g. OBEX protocol) that use serial line as the transport mechanism.

15 On top of the link and transport protocols, the applications still need some specific protocols to complete the protocol stack. In the Bluetooth architecture, the application-specific protocols are added on top of RFCOMM or directly on the L2CAP. L2CAP can only be accessed via a protocol which is supported by a Bluetooth profile such as RFCOMM..

20

The enumerated application-specific protocols offer the basic functionality in the Bluetooth environment and they provide only the cable-replacement capabilities. Features such as broadcasting, point-to-multipoint topologies, and scatternet possibilities are not really utilised by these current high-level protocols and usage models. Thus, there are numerous possibilities for developers to create more applications, the nature of which can be totally different from the existing ones.

The object of the present invention is to achieve a wireless communication between a processing unit and a printer using a safe transmission and an increased transmission range compared to the infrared transmission used in the above mentioned US-patent.

30

SUMMARY OF THE INVENTION

- 5 The object of the invention is to unravel the above mentioned drawbacks and achieve a way of printing a document in a data communications system using a protocol profiled for printing in the Bluetooth protocol architecture.

This is achieved according to the method and arrangement set forth in the
10 characterising parts of the independent claims.

Preferred embodiments are set forth in the independent claims.

- An advantage of the method and arrangement according to the present invention
15 is that it is possible to communicate wirelessly with a printer at a wide range, up to 10 meters and extendable up to 100 meters.

Another advantage is that it offers a safe transferring of data.

- 20 Yet another advantage is that the present invention makes it possible to wirelessly select a printer among available printers.

BRIEF DESCRIPTION OF THE DRAWINGS

25

- Figure 1 is a diagram showing the relationship between timeslots and frequency hops in a system using Bluetooth.
Figure 2 is a diagram illustrating the different function blocks of a Bluetooth system.
30 Figure 3 is a diagram showing the Bluetooth protocol stack.
Figure 4 is a schematic block diagram showing a communications system according to the present invention.

- Figure 5 is a schematic block diagram showing an entity according to the present invention.
- Figure 6 is a schematic block diagram showing a printer entity according to the present invention.
- 5 Figure 7 shows a flowchart of the method according to the invention.
- Figure 8 is a bloc diagram depicting a protocol overview over the Bluetooth protocols according to the invention.
- Figure 9 shows a signalling sequence over a typical SDP transaction.
- Figure 10 shows a signalling sequence over typical WPP transactions.
- 10 Figure 11 shows a signalling sequence over typical WPP transactions.
- Figure 12 shows a signalling sequence over typical WPP transactions.
- Figure 13 shows a signalling sequence over typical WPP transactions.
- Figure 14 shows a signalling sequence over typical WPP transactions.
- Figure 15 shows a signalling sequence over typical WPP transactions.

15

DESCRIPTION OF PREFERRED EMBODIMENTS

Figures 1-3 are related to prior art and described above under "Description of related art".

20

The wording "client" is in this disclosure defined as the entity sending a request, and the wording "server", is in this disclosure defined as the entity receiving a request.

- 25 Figure 4 shows a possible scenario of the present invention. A Bluetooth data communications system 401 includes two nodes whereof one is a processing unit, which in this example is a PC 402 and the other is a printer 403. A wireless printer protocol according to the invention is implemented in the Bluetooth protocol stack which is included in a entity, e.g. a PC-card 404, connected to or
- 30 implemented in the PC 402, and in a printer entity , e.g. a printer adapter 405, connected to or implemented in the printer 403. According to the Bluetooth standard the distance between the processing unit and the printer is up to 10

4
meters and extendable up to 100 meters. The printer adapter 405 might be connected to the printer port on the printer. The PC 402 and the printer 403 are connected to each other via a Bluetooth air interface 406. Both entities 404 and 405 comprise a respective computer, each computer comprising an internal
5 memory for storing computer program not visible in Figure 4.

The entity 404 connected to or implemented in the processing unit 402, will now be described more in detail. The entity, now referred to as 501 is shown in Figure
10 5. The entity 501 includes a Bluetooth protocol stack in which protocol stack a wireless printer protocol is implemented. The printer protocol comprises a printer client which communicates with a printer server by means of the wireless printer protocol, the Bluetooth protocol stack and air interface. The printer server is included in a printer but is not visible in Figure 5.

15 The entity 501 includes establishing means 502 for establishing a bi-directional wireless ACL connection between the processing unit and the printer by means of the Bluetooth protocol.

20 The entity 501 comprises further means 503 for sending a connection request message to the printer server and negotiating means 504 for negotiating configuration parameters with the printer server. The negotiating means 504 comprises sending means 505 for sending, to the printer server, a configuration request message including no new options if the printer client uses default values.
25 The negotiating means 504 comprises also sending means 506 for sending, to the printer server, a configuration request message including a suggestion of configuration options. The negotiating means 504 comprises further sending means 507 for sending, to the printer server, a further configuration request message including a suggestion of configuration options which differs from earlier
30 suggestions of configuration options. This latter sending means 507 is to be used if the printer client receives a response message from the printer server that the

configuration request was not acceptable due to e.g. unacceptable parameters, unknown option etc.

The entity 501 comprises sending means 508 for sending a set attribute request message to the printer server, the message comprising e.g. a coding table
5 concerning a negotiated coding type and is to be loaded by the printer server.

The entity 501 comprises sending means 509 for sending keep alive messages frequently to the printer server.

A keep alive timer 510 is implemented in the entity 501 and comprises starting
10 means 511 for starting and restarting the keep alive timer 510 each time a valid message is sent to the printer server and each time a valid message is received from the printer server. The keep alive timer 510 further comprises closing means 512 for closing the connection between to the printer server, when the keep alive timer 510 expires.

15

For starting one or more printjobs the entity 501 comprises starting means 513 which starting means 513 comprises sending means 514 for sending a request message to the printer server comprising a request to start a printjob.

20 The print data that is to be printed by the printer is sent by means of the sending means 515 for sending the print data to the printer server. Said means 515 includes sending means 516 for sending a number of request messages to the printer server, the messages comprising print data.

25 A printing process might be broken, e.g. because the printer runs out of paper or the ACL connection is broken, etc. This is reported by the printer server in a message received by the printer client. The entity 501 comprises means 527 for interpret the message and give a note to the user of the processing unit, e.g. by presenting the note on the screen of the PC.

30

E.g. a refill of paper or a new creation of a disconnected ACL connection might make, but the entity 501 comprises continuing means 517 for continuing the

printing process by continuing to send print data request messages to the printer server, starting with the print data subsequent to a last received print data acknowledgement message.

- 5 The entity 501 comprises stopping means 518 for stopping the keep alive timer 510 when an ACL connection is disconnected during a printing process.

The entity 501 further comprises requesting means 519 for requesting a reconnection of a session defined by the session identifier in a message sent to the
10 printer server to be used when a new ACL connection is created to the printer, after a break.

The entity 501 comprises stopping means 520 for stopping the print job said
stopping means 520 comprises sending means 521 for sending a message to the
15 printer server, the message comprising a request to stop the printjob. The stopping means 520 will be used when all data to be printed in a printjob is sent to the printer.

The entity 501 further comprises closing means 522 for closing the connection
20 between the processing unit and the printer, the closing means comprising sending means 523 for sending a message to the printer server, the message comprising a request to disconnect a session identified by a session identity.

The entity 501 comprises stopping means 524 for stopping the sending of keep
25 alive messages after closing a connection between the printer client and the printer server.

The entity also comprises a receiver 525 for receiving messages sent from a printer and a transmitter 526 for sending messages to the printer.

The printer entity 405 connected to or implemented in the printer 403 shown in Figure 4, will now be described more in detail. The printer entity, now referred to as 601 is shown in Figure 6. The printer entity 601, including a Bluetooth protocol stack in which a wireless printer protocol is implemented, said protocol
5 comprising a printer server which communicates, by means of the wireless printer protocol, the Bluetooth protocol stack and air interface, with a printer client, e.g. the printer client in the entity 501 described above. The printer client is included in a processing unit 402 and is not visible in Figure 6.

10 The printer entity 601 comprises a receiver 602 for receiving messages sent from a processing unit and a transmitter 603 for sending messages to the processing unit.

The printer entity 601 further comprises responding means 604 for responding upon a connection request whether the connection is successful or not, in a
15 response message sent to the printer client.

The printer entity 601 comprises negotiating means 605 for negotiating configuration parameters with the printer client within the processing unit. The negotiating means 605 comprises responding means 606 for responding upon
20 a configuration request whether the configuration options in the configuration request are supported by the printer server or not.

The negotiating means 605 comprises loading means 607 for loading a coding table or other optional attributes sent from the printer client.

The negotiating means 605 further comprises sending means 608 for sending a
25 response, whether the loading of the coding table was successful or not, to the printer client.

The printer entity 601 comprises sending means 609 for sending keep alive messages frequently to the printer client.

30 A keep alive timer 610 is implemented in the printer server within the printer entity 601. The printer entity 601 comprises starting and restarting means 611 for

starting the keep alive timer each time a valid message is received from the printer client and each time a valid message is sent to the printer client.

The printer entity 601 comprises starting means 612 for starting a print job. The
5 starting means 612 comprises confirming means 613 for confirming a start printjob request message sent to the printer client

The printer entity 601 comprises receiving means 614 for receiving print data from the printer client. The receiving means 614 including sending means 615 for
10 sending an acknowledgement message to the printer client after receiving a previous decided number of print data request messages.

The printer entity 601 comprises indicating means 616 for indicating, in a message sent to the printer client, that the printer has reported an exemption condition, e.g.
15 that the printer is out of paper, if the printer runs out of paper.

The printer entity 601 further comprises indicating means 617 for indicating, in a message sent to the printer client, when the printer clears the exemption, e.g. that the printer is refilled, when the printer is refilled.

20 The printer entity 601 comprises stopping means 618 for stopping the keep alive timer when an ACL connection to the processing unit is disconnected during a printing process.

The printer entity 601 comprises sending means 619 for sending a response
25 message to the printer client, according to whether a reconnection request is granted or not.

The printer entity 601 comprises stopping means 620 for stopping the print job. The stopping means 620 including sending means 621 for sending a response
30 message, after the printer server has received a request to stop the printjob, the message comprising a confirmation that this is apprehended and is sent to the printer client.

The printer entity 601 comprises sending means 622 for sending a response message to the printer client, according to whether a disconnection request is granted or not.

5

The printer entity 601 further comprises stopping means 623 for stopping the sending of keep alive messages after the connection to the printer client is closed.

- 10 **Figure 7** shows a flowchart of a possible scenario of the printing process according to the present invention.

The method includes the following steps:

15

701. A bi-directional wireless Asynchronous Connection-Less (ACL) connection is established between the processing unit 402 and the printer 403 by means of the printer protocol calling the L2CAP requesting the connection and the L2CAP creating the connection.
- 20 702. A connection is established between the printer client and the printer server for one or more printjobs.
703. The processing unit 402 and the printer 403 negotiate configuration parameters for said connection.
704. Keep alive messages are sent frequently during the session from the processing unit 402 to the printer 403 and from the printer 403 to the processing unit 402.
- 25 705. The processing unit 402 starts the printjob and
706. sends the printer data to the printer 403.
707. The print job is stopped and
- 30 708. the connection is closed between the processing unit 402 and the printer 403.

The method is implemented by means of a computer program product comprising the software code portions for performing the steps of the method. The computer program product is run on a computer stored in a digital computer within the process unit 402 and within the printer 403, e.g. in the printer adapter 405.

- 5 The computer program is loaded directly or from a computer usable medium, such as floppy-disc, CD, Internet etc.

Figure 8 is a bloc diagram depicting a protocol overview over the Bluetooth protocols including the wireless printer protocol **WPP** according to the invention. The left side represents the PC 801 and the right side represents the Printer 802. The Host Control Interface **HCI** is marked as a horizontal line. The **HCI** provides a command interface to the baseband controller, link manager, and access to hardware status and control registers.

- 15 **SDP**, **L2CAP** and **LMP** are described above, under Related Art. **WPP** will be described more in detail below.

The interface between two entities on the same layer, a so-called horizontal interface, is defined by its protocol 803, 804, 805 and 812, e.g. **L2CAP** on PC communicates with **L2CAP** on printer using the **L2CAP** protocol.

20

The actual flow of data (Protocol Data Units, PDU:s) is done between entities in different layers 806, 807, 808, 809, 810 and 811, a so-called vertical interface.

On the PC side the protocols is implemented by following applications:

- 25
- Client L2CA Application implements **L2CAP**
 - Client Printer Application implements **WPP**
 - Client Discovery Application implements **SDP**

On the printer side the protocols is implemented by following applications:

- 30
- Server L2CA Application implements **L2CAP**.
 - Server Printer Application implements **WPP**.

- Server Discovery Application implements SDP.

5 The printing method according to the invention will now be described more in detail.

A processing unit requires to print a document, i.e. to perform a printjob, by means of a printer.

10 The processing unit wishes to know which printers that are available, and select one of them, therefore the printing process starts with the Device Discovery procedure, which is a procedure known from the art. Figure 9 shows a sequence diagram of a typical SDP transaction between the Client Discovery Application 901 and the Server Discovery Application 902. It is assumed that inquire has been performed. As a result of inquire the class of device is retrieved. Class of device
15 indicates the type of device and which type of services the device supports. It is also assumed that a point to point connection with the server has been established, using L2CAP. The PrinterServiceClassId is represented as a Universally Unique Identifier (UUID) and is known by client discovery application.

20 A message, e.g. a denoted *SDP_ServiceSearchReq* message 903 is sent, from Client to Server, to ask which services, in this case printers that are available. The server returns service records handles associated with the respective available printers, e.g. in a denoted *SDP_ServiceSearchRsp* message 904.

25 The printer service record database serves as a repository of discovery-related information. All of the information about a service that is maintained by an SDP server is contained in a single service record. The service record consists entirely of a list of attributes. A service record handle uniquely identifies each service record within the SDP server, according to Service Discovery Protocol, Bluetooth
30 Specification version 1.0 B concerning SDP and Appendix VIII, Bluetooth Assigned Numbers, Bluetooth Specification version 1.0 B concerning assigned numbers for predefined attributes and their identity.

The Client selects one of the available printers and requests for its attributes, e.g. the address of the printer, a in a message, e.g. a denoted *SDP_ServiceAttributeReq* message 905 using the service record handle. The attributes are returned in one or
5 more messages, e.g. denoted *SDP_ServiceAttributeRsp* messages 906.

The Client stores the received attributes and terminate the L2CAP connection

A bi-directional wireless asynchronous connection-less (ACL) connection is established (701) between the processing unit and the printer. This is achieved by
10 means of the printer protocol in the processing unit calling the L2CAP in the within the same unit, requesting the connection to the printer. The printer is connected e.g. by means of the printer address being one of the attributes received. The L2CAP creates the connection and notifies the created connection the printer protocol.

15

Figure 10 shows sequence diagrams of a typical WPP transactions concerning the connection operations between the WPP Client 1001 and the WPP Server 1002, according to the invention

20

A creation of a session between a client printer application (source) and a server printer application (destination) is to be requested, i.e. for establishing a connection for one or more printjobs. This is performed by sending a message, e.g. a denoted *WPP_Connection_Reg* message 1003, from the WPP client 1001 to
25 the WPP server 1002. This is shown in Figure 10. A status indication to the client printer application whether the connection was successful or not and making the session valid if successful is required. This is performed in a message by the WPP server 1002, e.g. in a denoted *WPP_Connection_Rsp* message 1004, also shown in Figure 10. This message also includes a session identity.

30

The next step of the printing process is the WPP negotiation procedure according to the invention. Figure 11 a-c and 12 shows sequence diagrams of a typical WPP transactions concerning the negotiation operations between the WPP Client 1001 and the WPP Server 1002, according to the invention.

5

After creating the session a configuration of the WPP server 1002 is required. Examples of configuration options are e.g. the number of print data request messages to be received by the printer before return a confirmation message, coding type and table size.

10 Figure 11a, b and c shows three different sub-scenarios of a successful negotiation of a coding type for data compression. A message, e.g. a denoted *WPP_Configuration_Req* message, is sent from the WPP client 1001 to WPP server 1002 to establish an initial logical link transmission contract between the WPP client 1001 and WPP server 1002 and to negotiate configuration parameters, e.g.
15 the coding type. In this example the WPP server 1002 supports the coding types hamming, table size = 80 (default) and huffman table size = 80. The three respective sub-scenarios may be a continuation of the connection scenario in Figure 10.

20 In the first sub-scenario, shown in Figure 11a, the WPP client 1001 uses default values, i.e. hamming, table size = 80 and accordingly the *WPP_Configuration_Req* message 1101 sent, from the WPP client 1001 to the WPP server 1002, includes no new options. Since that is a coding type that the WPP server 1002 supports, it responses success in a message, e.g. a denoted *WPP_Configuration_Rsp* message
25 1102.

Figure 11b shows the second sub-scenario in which the WPP client 1001 requests the WPP server 1002, in message, e.g. a denoted *WPP_Configuration_Req* message, if hamming, table size = 100 can be used 1103. This is not a coding type that the
30 WPP server 1002 supports and accordingly it responses in a message, e.g. a denoted *WPP_Configuration_Rsp* message 1104, failure and suggests that hamming, table size = 80 can be used. The WPP client 1001 supports also hamming, table

size = 80 and responses this to the WPP server 1002 in a message, e.g. a denoted *WPP_Configuration_Req* message 1105. The WPP server responses success in a message, e.g. a denoted *WPP_Configuration_Rsp* message 1106.

- 5 In the third scenario, shown in **Figure 11 c**, the WPP client 1001 suggests an coding type which is unknown for the printer, i.e. a coding type not supported by the printer, and a size = 100, in a message, e.g. a denoted *WPP_Configuration_Req* message, sent 1107 to the WPP server 1002. Since this coding type is unknown for the WPP server 1002, it responses in a message, e.g. a denoted
10 *WPP_Configuration_Rsp* message 1108 failure and that the coding type is unknown. The WPP client 1001 then tries another coding type that it supports, in this example huffman, size = 80, in a subsequent message, e.g. a denoted *WPP_Configuration_Req* message 1109 sent to the WPP server 1002. The WPP server 1002 supports huffman, size = 80 and accordingly it responses success and
15 confirms huffman, size = 80 in a message, e.g. a denoted *WPP_Configuration_Rsp* message that is sent 1110 to the WPP client 1001.

- After the configuration negotiation of coding type according to e.g. the scenarios depicted in Figures 11a-c, the WPP client 1001 requests to set an attribute which is
20 illustrated in **Figure 12**. The WPP client 1001 sends a coding table concerning the negotiated coding type in a message, e.g. a denoted *WPP_Set_Attribute_Req* message sent 1201 to the WPP server 1002. The WPP server loads the coding table to be used and confirms whether it was successful or failure in a message, e.g.
25 a denoted *WPP_Set_Attribute_Rsp* message 1202 sent to the WPP client 1001.

- The next step of the printing process is the WPP printing procedure. **Figure 13a-d, 14 and 15** shows sequence diagrams of a typical WPP transactions concerning
30 the printing operations, between the WPP client 1001 and the WPP server 1002, according to the invention.

Figures 13a-d shows a first sub-scenario of a successful printing of one print job. Figure 13a shows the procedure for sending keep alive messages.

When the connection has been established and negotiation has been performed, keep alive messages are to be sent, by the WPP client 1001, 1303 and WPP server 1002, 1304, frequently, e.g. once each 5 second, as an indication that the source is up and running. Such a message is a denoted *WPP_Keep_Alive* message. If a break occurs when printing, the printer will find out that, since it does not receive any more keep alive messages. The printer then terminates the printjob and can let other users in. A break can also occur on the printer side. There is also occasions when the printer or processing unit are hard loaded, sending keep alive messages just to tell the receiver that it still alive but it goes slowly at the moment. When a connection has been disconnected by WPP client ,WPP client 1001 and WPP server 1002 shall stop sending denoted *WPP_Keep_Alive* messages.

A WPP Keep Alive Timer is restarted each time a valid message is received from the remote endpoint. The timer is implemented on both client and server side. If the Keep Alive timer expires the remote endpoint is considered faulty and the connection is closed and higher level applications is notified. The Keep Alive Timer shall be stopped when a link is disconnected and restarted when a new link is established with the remote endpoint. If a new link is established within a reasonable time, e.g. 10 seconds, the printjob continues where broken. Each WPP message will trigger a restart of a WPP timer.

In Figure 13b a start of a printjob and sending of data to be printed is shown. The WPP client 1001 requests the WPP server 1002 to start a printjob in a denoted *WPP_Start_Print_Req* message 1305 s, which in turn confirmed by the WPP server (1002) in a denoted *WPP_Start_Print_Cfm* message 1306. The WPP client then requests the WPP server 1002 to print data included in a number of denoted *WPP_Print_Data_Req* messages 1307, 1308. A confirmation is to be sent after the WPP server 1002 has received a number N *WPP_Print_Data_Req* messages 1307, 1308. The value of N is negotiated during configuration e.g. N = 4. The acknowledgement is e.g. sent in a denoted *WPP_Print_Data_Ack* message 1309.

This procedure goes on until all data to be printed is received by the printer server.
I.e. until the last *WPP_Print_Data_Req* message 1310 is received.

- When all data to be printed is sent to the printer server the client requests the printer server to stop the printjob. This is shown in Figure 13c wherein the WPP client 1001 sends a denoted *WPP_End_Print_Req* message 1311 to the WPP server 1002. That this is apprehended by the printer server is reported e.g. in a denoted *WPP_End_Print_Rsp* message 1312 sent to the WPP client 1001.
- After performing one or more printjobs or if a break of the printjob is requested, the client requests a disconnection of a session defined by the session identifier. Depicted in Figure 13d, this request is performed by e.g. sending a denoted *WPP_Disconnect_Req* message 1313 from the WPP client 1001 to the WPP server 1002 and a response, whether the disconnection is granted or not, is sent in the opposite direction in a denoted *WPP_Disconnect_Rsp* message 1314.

When the session is disconnected the WPP client 1001 and the WPP server 1002 stops sending *WPP_Keep_Alive* messages.

- Figures 14 shows a second sub-scenario of a successful printing of one printjob when the printer is out of paper. Negotiation has been performed, a connection is established and keep alive messages are sent as described above though not visible in Figure 14. The WPP client 1001 has requested the WPP server 1002 to start the printjob in a message, e.g. a denoted *WPP_Start_Print_Req* message 1401, which is responded success to in a message, e.g. a denoted *WPP_Start_Print_Rsp* message 1402. When the WPP client 1001 has requested the WPP server 1002 to print data included in a number of messages, e.g. denoted *WPP_Print_Data_Req* messages 1403, 1404, being acknowledged by the WPP server 1002 in a message, e.g. a denoted *WPP_Print_Data_Ack* message 1405, the printer is out of paper. The printer server then has to report this to the client. This can be performed by the WPP server 1002 sending a message, e.g. a denoted *WPP_Status_Ind* message

1406, indicating that the printer is out of paper to the WPP client 1001. The message is interpreted by the wireless printer protocol and reported to the user of the processing unit, e.g. by presenting a note on the PC screen. The message is obtained by a user of the processing unit including the client, who refills the printer. The printer server then reports that the printer is refilled to the WPP client 1001 by sending a message, e.g. a denoted *WPP_Status_Ind* message 1407. The last received denoted *WPP_Print_Data_Ack* message 1405 defines where to continue the printing by sending messages, e.g. denoted *WPP_Print_Data_Req* messages 1408, 1409 from the WPP client 1001 to the WPP server 1002. The printer will throw data if already printed or if a part of it has been printed. The printing process then continues as described above.

Figures 15 shows a third sub-scenario of a successful printing of one printjob when the ACL connection is disconnected. Negotiation has been performed, a connection is established and keep alive messages are sent as described above though not visible in Figure 15. The WPP client 1001 has requested the WPP server 1002 to start the printjob in a message, e.g. a denoted *WPP_Start_Print_Req* message 1501, which is responded success to in a message, e.g. a denoted *WPP_Start_Print_Rsp* message 1502. When the WPP client 1001 has requested the WPP server 1002 to print data included in a number of messages, e.g. *WPP_Print_Data_Req* messages 1503, 1504, the ACL connection is disconnected, indicated by HCI. The Keep Alive Timer is stopped by the WPP client 1001.

A reconnection of the session is required because it is possible for another client to start a printjob during ACL-disconnected. A session identity is used to identify the different WPP entities. If another job is ongoing the server will not accept the reconnection. The time the server will wait for the reconnection has to be handled by a reconnection timer. If the timer times out the ongoing job will be flushed. After creating a new ACL-connection a reconnection of the session is requested. This can be performed by the WPP client 1001 by sending a message, e.g. a denoted *WPP_Reconnect_Req* message 1506 requesting a reconnection of the

session defined by the session identifier. A response according to whether the reconnection is granted or not is sent in a message, e.g. a denoted *WPP_Reconnect_Rsp* message 1507. In this example it is granted. The WPP Keep Alive timer is started again. The last received denoted *WPP_Print_Data_Ack* message 1505 defines where to continue the printing by sending messages, e.g. a *WPP_Print_Data_Req* messages 1507, 1508 from the WPP client 1001 to the WPP server 1002. The printer server will throw data if already printed or if the packet is detected to be a retransmission. The printing process then continues as described above.

10

The present invention is not limited to the above-described preferred
15 embodiments. Various alternatives, modifications and equivalents may be used. Therefore, the above embodiments should not be taken as limiting the scope of invention, which is defined by the appendant claims.

CLAIMS

1. A method for printing a document in a data communications system, the system including a processing unit including a printer client (1001) and a printer including a printer server (1002), the processing unit and the printer using for communication between each other a wireless printer protocol, the Bluetooth protocol stack and air interface, the Bluetooth protocol stack including a wireless printer protocol and a Logical Link Control and Adaptation Protocol (L2CAP), the method including the steps of:
 - 5 *establishing* (701) a bi-directional wireless asynchronous connection-less (ACL) connection between the processing unit and the printer by means of the printer protocol calling the L2CAP requesting the connection and the L2CAP creating the connection;
 - establishing* (702) a connection for one or more printjobs between the printer client (1001) and the printer server (1002);
 - 15 *negotiating* (703) configuration parameters between the printer client (1001) and the printer server (1002);
 - sending* (704) keep alive messages frequently from the printer client (1001) to the printer server (1002) and from the printer server (1002) to the printer client (1001);
 - 20 *starting* (705) a print job;
 - sending* (706) the print data from the processing unit to the printer;
 - stopping* (707) the print job; and
 - closing* (708) the connection between the processing unit and the printer.
- 25 2. The method according to claim 1, comprising the further step to be taken before the step of establishing (701) a bi-directional wireless ACL connection:
 - selecting* a printer among a number of available printers.
- 30 3. The method according to claim 2, wherein the step of selecting a printer is performed by using the Device Discovery Protocol.

4. The method according to any of the previous claims, comprising the further step to be taken before the step of establishing (701) a bi-directional wireless ACL connection:
obtaining an address of a printer.
5. The method according to claim 4, wherein the step of obtaining an address of a printer is performed by using the Device Discovery Protocol.
6. The method according to claim 5, wherein the establishing a connection for one or more printjobs is performed by sending a connection request message (1003) from the printer client (1001) to the printer server (1002).
7. The method according to claim 6, wherein the establishing a connection for one or more printjobs is performed by responding upon the request whether the connection was successful or not, in a response message (1004) sent from the printer server (1002) to the printer client (1001).
8. The method according to any of the previous claims, wherein the step of negotiating configuration parameters (503), between the printer client (1001) and the printer server (1002), is performed by the printer client (1001) requesting configuration in a message (1101) sent to the printer server (1002), the message including no new options, if printer client (1001) uses default values.
9. The method according to any of the previous claims, wherein the step of negotiating configuration parameters(503), between the printer client (1001) and the printer server (1002), is performed by the printer client (1001) requesting configuration in a message (1103) sent to the printer server (1002), the message including a suggestion of configuration options.
10. The method according to claim 9, wherein said configuration request is responded to by the printer server (1002) in a message (1102, 1104, 1106)

indicating whether the configuration options in the configuration request are supported by the printer server (1002) or not.

11. The method according to claim 10, including the further step, if the
5 configuration request responds failure;
sending a further configuration request message (1105, 1109) from the printer client (1001) to the printer server (1002), the message including suggestion of configuration options which differs from earlier suggestions of configuration options..
12. The method according to any of the previous claims, comprising the further
10 step to be taken after the step of negotiating configuration parameters (503):
sending a set attribute request message (1201) from the printer client (1001) to the printer server (1002) the message comprising a coding table concerning a negotiated coding type.
13. Method according to claim 12, comprising the further step of:
the printer server (1002) *loading* the coding table by means of said received set
20 attribute request message (1201).
14. Method according to claim 13, comprising the further step of:
sending a response whether the loading of the coding table was successful or not
in a message (1202) from the printer server (1002) to the printer client (1001).
15. The method according to any of the previous claims, wherein a keep alive
25 timer is implemented in the printer client (1001) and in the printer server (1002), comprising the further step of:
starting the keep alive timer each time a valid message is received from the remote endpoint.
16. The method according to claim 15, wherein said keep alive timer expires,
30 comprising the further step of:

closing the connection.

17. The method according to any of the previous claims, wherein the step of starting a print job (505) is performed by the printer client (1001) requests the printer server (1002) to start a printjob in a request message (1305).

18. The method according to claim 17, wherein said start printjob request message (1305) is received and confirmed by the printer server (1002), the confirmation sent in message (1306) to the printer client (1001).

10

19. The method according to any of the previous claims, wherein the step of sending the print data from the processing unit to the printer (506), is performed by requesting the printer server (1002) to print data sent in a number of messages (1307, 1308, 1310).

15

20. The method according to claim 19, comprising the further step to be taken after the printer server (1002) have received a previous decided number of print data request messages:

sending an acknowledgement message (1309) from the printer server (1002) to the printer client (1001).

20

21. The method according to any of the previous claims, comprising the further step to be taken if the printer runs out of paper:

indicating that the printer is out of paper in a message (1406) sent from the printer server (1002) to the printer client (1001).

25

22. The method according to claim 21, comprising the further step to be taken when the printer is refilled:

indicating that the printer is refilled in a message (1407) sent from the printer server (1002) to the printer client (1001).

30

23. The method according to claim 22, comprising the further step to be taken after the printer client (1001) has received an indication message (1407) that the printer is refilled:
continuing the printing process by continuing to send print data request messages, (1408, 1409) starting with the print data subsequent to the last received print data acknowledgement message (1405).
24. The method according to any of the claims, wherein the ACL connection is disconnected during printing, the method comprising the further step of:
stopping the keep alive timer.
25. The method according to claim 24, wherein a new ACL connection is created comprising the further step of:
requesting a reconnection of the session defined by the session identifier in a message (1506) sent from the printer client (1001) to the printer server (1002).
26. The method according to claim 25, comprising the further step of:
sending a response according to whether the reconnection is granted or not in a message (1507) from the printer server (1002) to the printer client (1001).
27. The method according to claim 26, comprising the further step to be taken after the printer client (1001) has received a granted reconnection response:
continuing the printing process by continuing to send print data request messages (1508, 1509), starting with the print data subsequent to the last received print data acknowledgement message (1505).
28. The method according to any of the previous claims, wherein the step of stopping the print job (707), is performed by, after sending all data to be printed to the printer server (1002), sending a request to stop the print job in a message (1311) from the printer client (1001) to the printer server (1002).

29. The method according to claim 28, comprising the further step to be taken
after the printer server (1002) has received a request to stop the printjob;
sending a response message (1312), comprising a confirmation that this is
apprehended, from the printer server (1002) to the printer client (1001).
5
30. The method according to any of the previous claims, wherein the step of
closing the connection between the processing unit and the printer (708) is
performed by the printer client (1001) requesting a disconnection of the
session defined by the session identity in a message (1313) sent to the printer
server (1002).
10
31. The method according to claim 30, wherein the printer server responds to
whether the disconnection was granted or not, in a response message (1314)
sent from the printer server (1002) to the printer client (1001).
15
32. The method according to any of the previous claims, comprising the further
step to be taken after the step of closing the connection between the
processing unit and the printer (708):
stopping the sending of keep alive messages.
20
33. A computer program product directly loadable into the internal memory of a
digital computer within a processing unit or printer in a communication
system, comprising the software code portions for performing the steps of any
of the claims 1-32, when said product is run on a computer.
25
34. A computer program product stored on a computer usable medium,
comprising readable program for causing a computer within a processing unit
or printer in a communication system, to control an execution of the steps of
any of the claims 1-32.
30
35. An entity (501) included in a Processing unit (402), the entity includes a
Bluetooth protocol stack comprising a Logical Link Control and Adaptation

- Protocol (L2CAP) characterised in that the Bluetooth protocol stack further comprises a wireless printer protocol, said printer protocol comprising a printer client which communicates (803) with a printer server, included in a printer (403), by means of the Bluetooth protocol stack and air interface, the entity (501) further comprises:
- establishing means (502) for establishing a bi-directional wireless ACL connection to the printer (403) by calling the L2CAP requesting the connection;
 - establishing means (503) for establishing a connection for one or more printjobs
 - negotiating means (504) for negotiating configuration parameters with a printer server within the printer(403);
 - sending means (509) for sending keep alive messages frequently to the printer server ;
 - starting means (513) for starting a print job;
 - sending means (515) for sending the print data to the printer server;
 - stopping means (520) for stopping the print job; and
 - closing means (522) for closing the connection between the processing unit (402) and the printer (403).
36. The entity (501) according to claim 35, characterised by comprising sending means for sending a connection request message from the printer client to the printer server.
37. The entity (501) according to any of the claims 35-36 wherein when negotiating configuration parameters, the printer client uses default values, characterised by comprising sending means (505) for sending a configuration request message to the printer server, the message including no new options.
38. The entity (501) according to any of the claims 35-37, characterised by comprising sending means (506) for sending a configuration request message

to the printer server, the message including a suggestion of configuration options.

39. The entity (501) according to any of the claims 35-38, characterised by
5 comprising sending means (507) for sending a further configuration request to the printer server, the message including suggestion of configuration options which differs from earlier suggestions of configuration options.
40. The entity (501) according to any of the claims 35-39, characterised by
10 comprising sending means (508) for sending a set attribute request message to the printer server, the message comprising a coding table concerning a negotiated coding type.
41. The entity (501) according to any of the claims 35-40, characterised in that a
15 keep alive timer (510) is implemented in the printer client.
42. The entity (501) according to claim 41, characterised by comprising starting
means (511) for starting the keep alive timer (510) each time a valid message is
received from the printer (403).
- 20 43. The entity (501) according to claim 42, characterised by comprising closing
means (512) for closing the connection between the printer client and the
printer server, when the keep alive timer(510) expires.
- 25 44. The entity (501) according to any of the claims 35-43, characterised by
comprising sending means (514) for sending a request message to the printer
server comprising a request to start a printjob.
45. The entity (501) according to any of the claims 35-44, characterised by
30 comprising sending means (516) for sending a number of request messages to
the printer server, the messages comprising print data.

46. The entity (501) according to any of the claims 35-45, wherein a refill of paper has broken a printing process, **characterised** by comprising continuing means (517) for continuing the printing process by continuing to send print data request messages to the printer server, **starting** with the print data subsequent to the last received print data acknowledgement message.
47. The entity (501) according to claim 41, **characterised** by comprising stopping means (518) for stopping the keep alive timer when the ACL connection is disconnected during a printing process.
48. The entity (501) according to any of the claims 35-47, wherein a new ACL connection is created to the printer after a break, **characterised** by comprising requesting means (519) for requesting a reconnection of a session defined by the session identifier in a message sent to the printer server.
49. The entity (501) according to claim 48, wherein a granted reconnection response message is received, **characterised** by comprising continuing means (517) for continuing the printing process by continuing to send print data request messages to the printer server, **starting** with the print data subsequent to the last received print data acknowledgement message.
50. The entity (501) according to any of the claims 35-49, wherein all data to be printed is sent to the printer **characterised** by comprising sending means (521) for sending a message to the printer server, the message comprising a request to stop the printjob.
51. The entity (501) according to any of the claims 35-50, **characterised** by comprising sending means (523) for sending a message to the printer server, the message comprising a request to disconnect a session identified by a session identity.

52. The entity (501) according to any of the claims 35-51, characterised by comprising stopping means (524) for stopping the sending of keep alive messages after closing a connection between the printer client and the printer server.

5

53. A printer entity (601) included in a Printer (403), the printer entity (601) including a Bluetooth protocol stack comprising a Logical Link Control and Adaptation Protocol (L2CAP) characterised in that the Bluetooth protocol stack further includes a wireless printer protocol, said printer protocol comprising a printer server which communicates with a printer client, included in a processing unit (402), by means of the wireless printer protocol, the Bluetooth protocol stack and air interface, the printer entity (601) further comprises:

10

negotiating means (605) for negotiating configuration parameters with a printer client within the processing unit;

15

sending means (609) for sending keep alive messages frequently to the printer client ;

starting means (612) for starting a print job;

receiving means (614) for receiving print data from the printer client; and

20

stopping means (620) for stopping the print job.

54. The printer entity (601) according to claim 53 characterised in comprising responding means (604) for responding upon a connection request whether the connection is successful or not, in a response message sent to the printer client.

25

55. The printer entity (601) according to any of the claims 53-54 characterised in comprising responding means (606) for responding upon a configuration request whether the configuration options in the configuration request are supported by the printer server or not.

30

56. The printer entity (601) according to any of the claims 53-55 characterised in comprising loading means (607) for loading a coding table sent from the printer client.
- 5 57. The printer entity (601) according claim 56 characterised in comprising sending means (608) for sending a response whether the loading of the coding table was successful or not to the printer client.
58. The printer entity (601) according to any of the claims 53-57, characterised in
10 that a keep alive timer (610) is implemented in the printer server.
59. The printer entity (601) according to claim 58, characterised by comprising starting means (611) for starting the keep alive timer each time a valid message is received from the printer.
- 15 60. The printer entity (601) according to any of the claims 53-59, characterised in comprising confirming means (613) for confirming a start printjob request message sent to the printer client.
- 20 61. The printer entity (601) according to any of the claims 53-60, characterised in comprising sending means (615) for sending an acknowledgement message to the printer client after receiving a previous decided number of print data request messages.
- 25 62. The printer entity (601) according to any of the claims 53-61, characterised in comprising indicating means (616) for indicating, in a message sent to the printer client, that the printer is out of paper, if the printer runs out of paper.
- 30 63. The printer entity (601) according to any of the claims 53-62, characterised in comprising indicating means (617) for indicating, in a message sent to the printer client, that the printer is refilled, when the printer is refilled.

64. The printer entity (601) according to any of the claims 53-63, characterised by comprising stopping means (618) for stopping the keep alive timer when an ACL connection to the processing unit is disconnected during a printing process.
- 5
65. The printer entity (601) according to any of the claims 53-64, characterised in comprising sending means (619) for sending a response message to the printer client, according to whether a reconnection request is granted or not.
- 10
66. The printer entity (601) according to any of the claims 53-65, characterised in comprising sending means (621) for sending a response message, after the printer server has received a request to stop the printjob, the message comprising a confirmation that this is apprehended and is sent to the printer client.
- 15
67. The printer entity (601) according to any of the claims 53-66, characterised in comprising sending means (622) for sending a response message to the printer client, according to whether a disconnection request is granted or not.
- 20
68. The printer entity (601) according to any of the claims 53-67, characterised in comprising stopping means (623) for stopping the sending of keep alive messages after the connection to the printer client is closed.
- 25
69. Communications system (401) characterised by comprising a processing unit (501) according to any of the claims 35-52 and a printer entity (601) according to any of the claims 53-68.

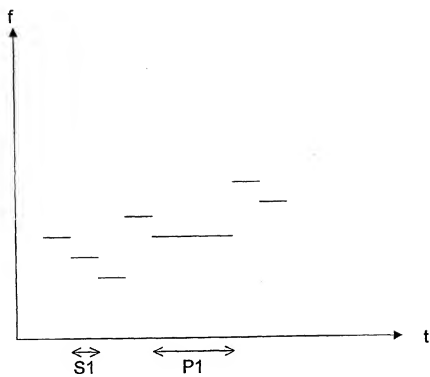


Fig. 1

2/13

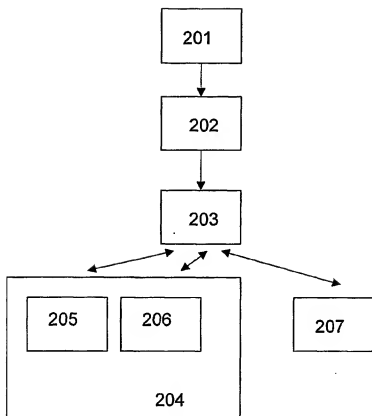


Fig. 2

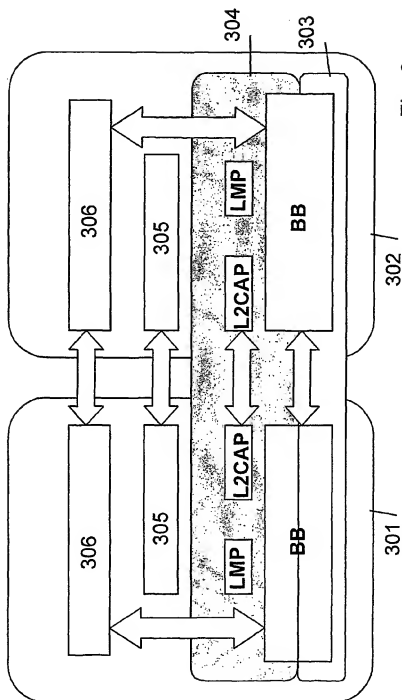


Fig. 3

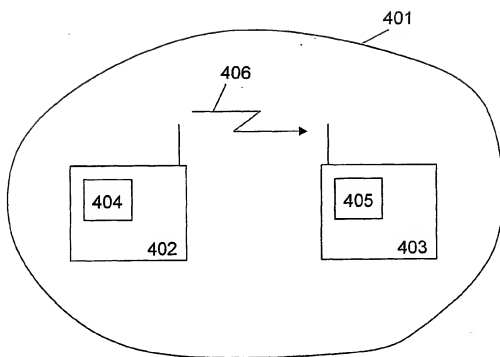


Fig. 4

5/13

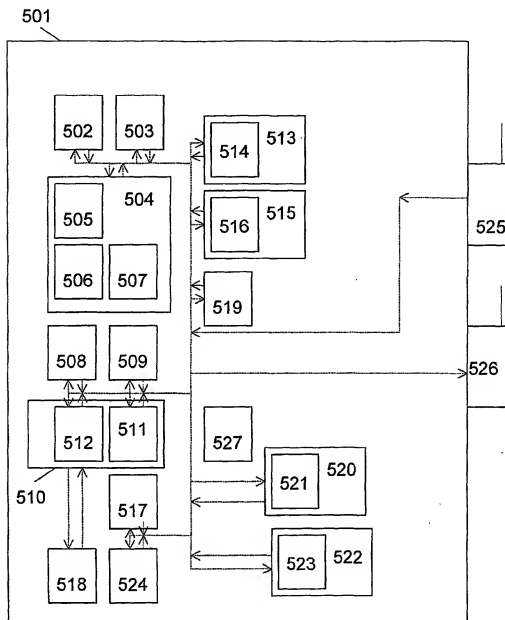


Fig. 5

6/13

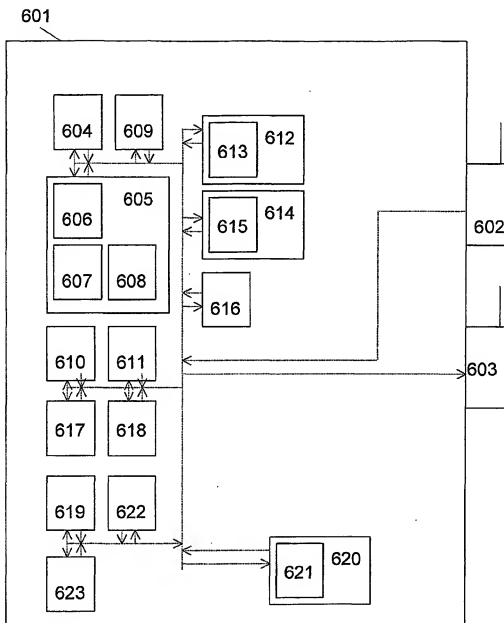


Fig. 6

7/13

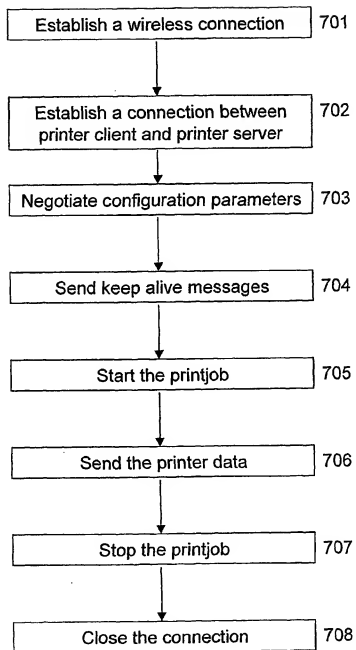


Fig. 7

8/13

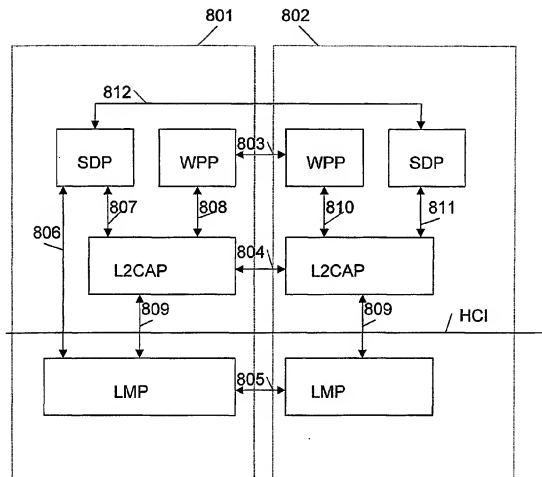


Fig. 8

9/13

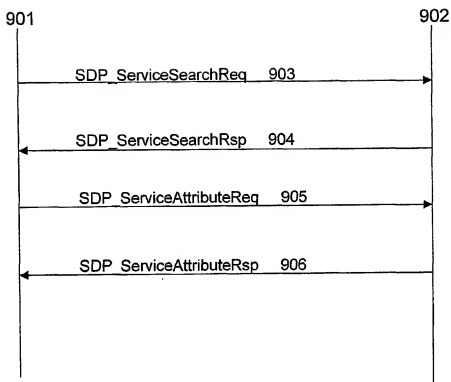
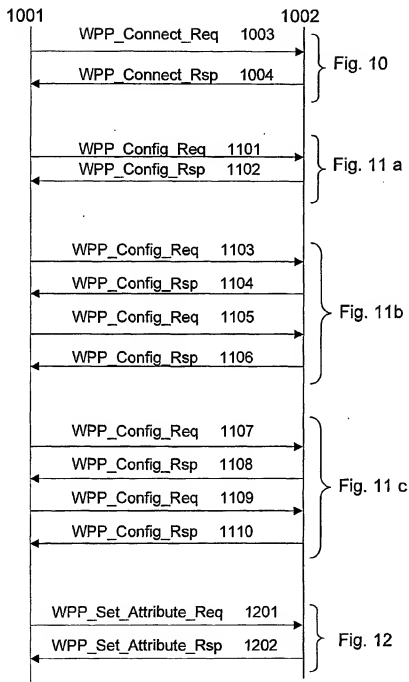
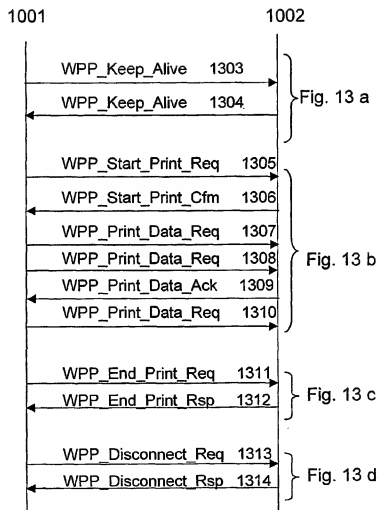


Fig. 9

10/13





12/13

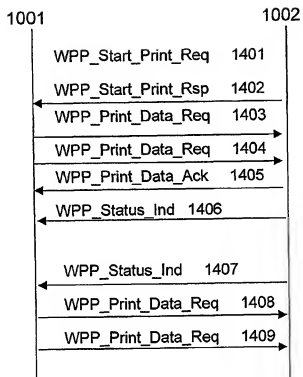


Fig. 14

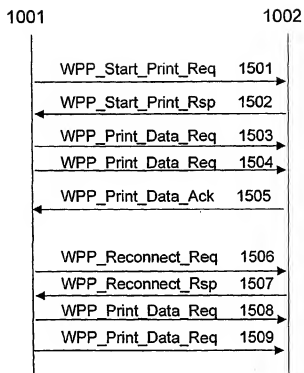


Fig. 15

INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 01/01094

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04L 12/58, H04Q 7/22

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: H04L, G06F, H04M, H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI DATA, EPO-INTERNAL

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 9941876 A1 (TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)), 19 August 1999 (19.08.99), see whole document	1,33-35,53, 69
Y	---	2-32,36-52, 54-68
X	EP 1001348 A2 (NOKIA MOBILE PHONES LTD), 17 May 2000 (17.05.00), see whole document	1,33-35,53, 69
Y	---	2-32,36-52, 54-68

☒ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

7 Sept 2001

Date of mailing of the international search report

25-09-2001

Name and mailing address of the ISA/
Swedish Patent Office
Box 5055, S-102 42 STOCKHOLM
Facsimile No. +46 8 666 02 86

Authorized officer

Roger Bou Faisal/LR
Telephone No. +46 8 782 25 00

INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 01/01094

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	METTALA,Riku: Bluetooth Architecture, Version 1,0; Aug. 25th 1999	1,33-35,53, 69
Y	---	2-32,36-52, 54-68
A	HAARTSEN,Jaap: Bluetooth - The universal radio interface for ad hoc, wireless connectivity. Ericsson Review No.3, 1998	1,33-35,53, 69
Y	---	2-32,36-52, 54-68
A	US 6055062 A (DINA,D. ET AL.), 25 April 2000 (25.04.00), cited in Application,see whole document	1-69
A	WO 0021203 A1 (COMSENSE TECHNOLOGIES, LTD), 13 April 2000 (13.04.00), page 6, line 26 - page 7, line 3; page 23, line 13 - line 30, abstract	1,33-35,53, 69
A	EP 1003307 A2 (RICOH COMPANY), 24 May 2000 (24.05.00), column 10, line 42 - column 11, line 23; column 13, line 17 - line 39	1,33-35,53, 69

INTERNATIONAL SEARCH REPORT
Information on patent family members

02/08/01

International application No.

PCT/SE 01/01094

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9941876 A1	19/08/99	AU 2650199 A BR 9907826 A CN 1290438 T EP 1055307 A	30/08/99 24/10/00 04/04/01 29/11/00
EP 1001348 A2	17/05/00	FI 982353 A JP 2000155727 A	30/04/00 06/06/00
US 6055062 A	25/04/00	GB 2332878 A,B GB 9827838 D	07/07/99 00/00/00
WO 0021203 A1	13/04/00	AU 2455700 A AU 5881999 A AU 5996899 A AU 5997299 A EP 1116155 A WO 0021020 A WO 0029920 A AU 5441899 A EP 1108224 A	05/06/00 03/04/00 26/04/00 26/04/00 18/07/01 13/04/00 25/05/00 21/03/00 20/06/01
EP 1003307 A2	24/05/00	JP 2000162926 A	16/06/00